

SVEUČILIŠTE U ZAGREBU  
FILOZOFSKI FAKULTET  
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI  
Ak. god. 2016./ 2017.

Valentino Munić

**PRIMJER INFORMACIJSKOG SUSTAVA U MREŽNOM  
OKRUŽENJU**

**Aplikacija za pomoć učenicima pri upisu u srednju školu**

Završni rad

Mentor: dr.sc. Kristina Kocijan, doc.

ZAGREB, 2017.

## Sadržaj

1.	UVOD .....	3
2.	OSNOVNI POJMOVI .....	4
2.1.	Informacijski sustav .....	4
2.2.	Web 2.0 okruženje .....	4
2.3.	NISpuSŠ.....	5
2.4.	HTML .....	8
2.5.	CSS .....	10
2.6.	Javascript.....	11
2.7.	Google Maps API .....	11
2.8.	Bootstrap .....	12
3.	PRIPREMA ZA IZRADU WEB SJEDIŠTA .....	13
3.1.	Ideja i svrha projekta.....	13
3.2.	Izrada baze podataka.....	14
3.3.	Struktura web sjedišta .....	15
4.	IZRADA WEB SJEDIŠTA.....	16
4.1.	HTML struktura web sjedišta .....	16
4.2.	CSS struktura web sjedišta.....	19
4.3.	Kreiranje Google Maps JavaScript karte .....	21
5.	ZAKLJUČAK.....	26
6.	LITERATURA .....	27

## Sažetak

U ovom radu opisan je postupak izrade informacijskog sustava u mrežnom okruženju, na primjeru izrade interaktivne karte. Postupak izrade opisan je od početne ideje projekta, preko pripreme za izradu web sjedišta, do samog projektiranja aplikacije. Rad obuhvaća korake u izradi web sjedišta od nastanka ideje, svrhe projekta, izrade web sjedišta, njegove HTML i CSS strukture te izrade interaktivne karte pomoću Google Maps JavaScript API.

**Ključne riječi:** *HTML, CSS, JavaScript, NISpuSŠ, Google Maps API*

## **An example of an information system in a network environment: application for assistance with high school enrollment**

### **Abstract**

This paper describes the process of creating an information system in a network environment, on an example of an interactive map. The process of creating the system is described from the initial idea of the project and preparations for building a website, to writing the application itself. The paper includes steps in process of building a website, from the basic idea and purpose of the project, to building the website, it's HTML and CSS structure and creating an interactive map using the Google Maps JavaScript API.

**Key words:** *HTML, CSS, JavaScript, NISpuSŠ, Google Maps API*

## 1. Uvod

U ovom radu pisat ću o postupku izrade i dizajniranja informacijskog sustava u mrežnom okruženju: aplikacije za pomoć učenicima pri upisu u srednju školu. Aplikacija na interaktivnoj karti vizualno prikazuje lokacije srednjih škola u gradu Zagrebu.

Rad je temeljen na primjeru mrežne stranice koja na interaktivnoj karti vizualno prikazuje lokacije srednjih škola u gradu Zagrebu. Prikaz se može filtrirati prema vrsti obrazovnih programa, na gimnazijske i strukovne te prema samim nazivima obrazovnih programa. Stranica je dostupna na poveznici: <http://ikzstudentskiprojekti.ffzg.hr/UpisiHr/izbornik.html>.

Na izradu ovog projekta motivirao me moj rad u Hrvatskoj akademskoj i istraživačkoj mreži (CARNet), u službi Podrške obrazovnom sustavu, u sklopu koje sam pružao podršku učenicima, roditeljima i nastavnicima u vrijeme upisa u srednje škole. U svakodnevnom kontaktu s korisnicima upoznao sam se s njihovim potrebama te s prednostima i nedostacima Nacionalnog informacijskog sustava prijava i upisa u srednje škole<sup>1</sup> koje ću detaljnije objasniti u nastavku rada.

U drugom poglavlju rada navest ću i objasniti neke osnovne pojmove i alate korištene pri izradi ove aplikacije. U trećem poglavlju opisat ću planiranje izrade projekta i izradu baze podataka. U trećem poglavlju ću također opisati ideju i svrhu ovog projekta te objasniti moju motivaciju za izradu aplikacije. Postupak izrade aplikacije i njenu informacijsku strukturu opisat ću u četvrtom poglavlju.

---

<sup>1</sup> U daljnjem tekstu „NISpuSŠ“.

## 2. Osnovni pojmovi

U ovom ću poglavlju navesti neke od osnovnih pojmova i alata koje sam koristio pri izradi ovog projekta. Objasniti ću njihove funkcije te navesti primjere u kojima sam ih koristio prilikom izrade ovog projekta.

### 2.1. Informacijski sustav

*„Ono što razlikuje informacijski od ostalih sustava i djelatnosti jest osebujnost informacijske djelatnosti, specifičnost njezinih normi, funkcija i odnosa. Informacijski je sustav mreža informacijskih djelatnosti, službi i odnosa; kao takva, ta je mreža inkorporirana u sva društvena područja i djelatnosti i tvori društvenu infrastrukturu. Kao nešto različito od drugih društvenih djelatnosti, podsustava ili sustava, informacijski je sustav to tek po svojim funkcijama, organizaciji i korisnicima.“* (Tuđman et al, 1991).

Informacijski sustav prikuplja, pohranjuje, čuva, obrađuje i isporučuje potrebne informacije tako da su one dostupne svim njegovim korisnicima. (Radovan, 1991). U ovom projektu informacijski sustav služi za pohranjivanje podataka o srednjim školama i srednjoškolskim obrazovnim programima te ih na zahtjev prikazuje u strukturiranom obliku na interaktivnoj zemljopisnoj karti.

### 2.2. Web 2.0 okruženje

Termin Web 2.0 popularizirali su Tim O'Reilly i Dale Dougherty na O'Reilly Media Web 2.0 konferenciji 2004. godine. Termin je od tada prešao u široku uporabu, iako se sam termin po prvi puta spominje u članku „Podijeljena budućnost“ (*„Fragmented future“*) iz 1999. godine, autora Darcyja DeNuccija. Sam termin ne odnosi

se na nadogradnju tehničkih specifikacija, već na način na koji mrežne stranice i aplikacije nastaju i koriste se.

Web 2.0 stranica može omogućiti korisnicima da međusobnom interakcijom i suradnjom stvaraju sadržaj na webu, i virtualnoj zajednici, za razliku od prve generacije weba gdje je korištenje sadržaja bilo ograničeno na njegovo pregledavanje. Neke od osobina Weba 2.0 su društvene mreže (kao što su Facebook i Twitter), blogovi, „wikiji“, platforme za dijeljenje i gledanje video sadržaja i drugo.

Glavna karakteristika Weba 2.0 je da se web koristi kao platforma. Korisnici aplikacije mogu koristiti u potpunosti kroz preglednik, korisnicima pripada sadržaj i nad njime imaju kontrolu, a sama arhitektura weba korisnike potiče da sudjeluju u stvaranju i dijeljenju sadržaja (CARNet, 2009).

Primjer Web 2.0 tehnologije je i Google Maps API koji je u ovom projektu korišten za izradu interaktivne karte. Razvojni programeri mogu pomoću tehnologije koju je razvio Google razvijati složene sustave navigacije, geolokacije i mapiranja, koje njihovi korisnici mogu dodatno prilagoditi svojim potrebama. U ovom projektu korisnici mogu utjecati na prikazani sadržaj filtriranjem lokacija pomoću padajućeg izbornika.

## 2.3. NISpuSŠ

*„Nacionalni informacijski sustav prijava i upisa u srednje škole jest središnji informacijsko-administracijski servis putem kojeg učenici koji planiraju nastavak školovanja u srednjoškolskim ustanovama prijavljuju upis u obrazovne programe, odnosno srednju školu.“ (MZO, 2017).*

Sustav NISpuSŠ dio je projekta elektroničkih upisa u srednje škole. Projekt e-upisa pokrenut je kako bi se upisi u srednju školu učinili jednostavnijima, transparentnijima i učinkovitijima. Učenici sustavu pristupaju putem mrežne stranice <http://www.upisi.hr/>, koja je u rad puštena 15. travnja 2013. godine, što je i službeno označilo početak projekta e-upisa (CARNet, 2013).

Sustav su izradili CARNet i Ministarstvo obrazovanja u suradnji s Fakultetom elektrotehnike i računalstva i tvrtkom KING ICT. Postupak e-upisa reguliran je Pravilnikom o elementima i kriterijima za izbor kandidata za upis u I. razred srednje škole te Pravilnikom o izmjenama i dopunama spomenutog Pravilnika. Sustav NISpuSŠ kolokvijalno se još naziva i Nispuš, Upisi Haer ili e-upisi.

Sustav NISpuSŠ pokriva cijeli proces upisa učenika u srednje škole, od pretraživanja obrazovnih programa prema željenim kriterijima, prijava odabranih programa te uvida u rezultate po tim programima, do ostvarivanja prava na upis u srednju školu. Iako postoje iznimke, u pravilu je moguće prijaviti do šest obrazovnih programa, na način da se lista prioriteta slaže od programa koji učenik najviše želi upisati, prema onima koje želi upisati manje. Lista odabranih obrazovnih programa zaključava se svim kandidatima na propisani datum. Trenutkom objave konačnih ljestvica poretka, kandidati stječu pravo upisa u obrazovni program svog najvišeg prioriteta, a za koji se nalaze unutar upisne kvote odobrene od strane Ministarstva znanosti i obrazovanja (MZO, 2017.)

Prednosti sustava NISpuSŠ u odnosu na prijašnji način prijava i upisa u srednje škole su očite; učenici mogu prijaviti do šest obrazovnih programa koje bi htjeli upisati, dok su ranije fizičku prijavu mogli obaviti samo u jednoj školi. Učenici imaju uvid u ljestvice poretka u gotovo realnom vremenu. Ljestvice poretka osvježavaju se svaki puni sat, dok je ranije nakon prijave bilo potrebno čekati konačne rezultate, ili neslužbene ukoliko ih je škola odlučila objaviti. Učenici mogu u bilo kojem trenutku odjavljivati prijavljene programe ili mijenjati listu prioriteta, dok je ranije, uz postojanje iznimaka, predajom dokumenata u željenoj školi njihova prijava bila konačna.

Uz sve navedene prednosti, sustav NISpuSŠ ima i neke nedostatke. Budući da je Pravilnik o upisima, s pripadajućim izmjenama i dopunama te brošurama s uputama, poprilično pozamašan, stručan i nepregledan, većina učenika i roditelja, pa i neki članovi upisnih povjerenstava, nikada ga ne pročitaju. To dovodi do situacija u kojima učenici ne razumiju koncept liste prioriteta i ljestvice poretka, pa prijavljuju škole koje zapravo ne žele upisati te budu nezadovoljni krajnjim rezultatom. Također, kako se ljestvice poretka

osvježavaju svaki puni sat, a zbog mogućnosti ljudske pogreške i naknadnih korekcija one nisu konačne u trenutku zaključavanja odabira programa, već to postaju objavom konačnih rezultata, učenici do objave rezultata ne mogu sa stopostotnom sigurnošću znati u koji će program ostvariti pravo upisa. Iako su dostupni dokumenti s navedenim minimalnim brojem bodova potrebnih za ostvarivanje prava upisa u pojedine programe u prethodnim godinama, a koji bi trebali služiti za orijentaciju pri odabiru obrazovnih programa, učenici za njih većinom ne znaju. Još jedna mana sustava NISpuSŠ je to što način pretraživanja, pregledavanja i odabira obrazovnih programa nekima može biti nejasan i nepregledan, kao što je prikazano na slici 1.

Vrsta programa

Program

Županije

Vrsta škole

Škola

Program	Škola	Mjesto	Broj mjesta	
Administrator (prilagođeni program)	Centar za odgoj i obrazovanje Vinko Bek Zagreb	Zagreb	7	<input type="button" value="Detalji"/>
Alatničar	Industrijska strojarska škola Zagreb	Zagreb	7	<input type="button" value="Detalji"/>
Autolakirer	Obrtnička i industrijska graditeljska škola Zagreb	Zagreb	1	<input type="button" value="Detalji"/>
Autolimar - JMO	Elektrostrojarska obrtnička škola Zagreb	Zagreb	5	<input type="button" value="Detalji"/>
Autolimar - JMO	Industrijska strojarska škola Zagreb	Zagreb	7	<input type="button" value="Detalji"/>
Autolimar (prilagođeni program)	Centar za odgoj i obrazovanje "Slava Raškaj" Zagreb Zagreb	Zagreb	4	<input type="button" value="Detalji"/>
Automehaničar - JMO	Industrijska strojarska škola Zagreb	Zagreb	3	<input type="button" value="Detalji"/>
Automehatroničar - JMO	Elektrostrojarska obrtnička škola Zagreb	Zagreb	1	<input type="button" value="Detalji"/>

Slika 1: Pretraživanje programa

Učenici iz padajućih izbornika trebaju odabrati kriterije pretraživanja, a u rezultatima pretraživanja pregledati opće informacije o školama, programima, te uvjetima



upisa. Tek na kraju prikaza informacija o programu nalazi se opcija za odabir programa, što je vidljivo na slici 2.<sup>2</sup>

Prikaz 14 od 20 Prva ekonomska škola Zagreb - Ekonomist

Dodatni bodovi:

Opis	Broj bodova	Ulazi u prag
Kandidat koji je osnovnu školu završio prema rješenju ureda državne uprave u županiji odnosno Gradskoga ureda za obrazovanje, kulturu i sport Grada Zagreba o primjerenom obliku školovanja	izravan upis *	Ne
Pripadnik romske nacionalne manjine koji živi u uvjetima koji su mogli utjecati na uspjeh u osnovnoj školi	2 *	Ne
Kandidat koji je prethodno obrazovanje završio po redovitome nastavnom planu i programu, a kojemu su teška zdravstvena oštećenja ili kronične bolesti i/ili dulje liječenje utjecale na postizanje rezultata tijekom prethodnog obrazovanja te značajno smanjuju mogućnost izbora srednjoškolskoga obrazovnog programa	1 *	Ne
Kandidat koji živi u otežanim uvjetima uzrokovanim ekonomskim, socijalnim te odgojnim čimbenicima koji su mogli utjecati na uspjeh u osnovnoj školi	1 *	Ne

\* Ako kandidat ostvaruje pravo na bodove temeljem više stavaka označenih zvjezdicom, za vrednovanje se uzima samo najpovoljnija vrijednost.

Stupac „Ulazi u prag“ označava računavaju li se dodatni bodovi u minimalni bodovni prag potreban za prijavu obrazovnog programa.

Napomena:

Zatvori Odaberi

Slika 2: Detalji i odabir programa

## 2.4. HTML

HyperText Markup Language (HTML) je skup oznaka koji se koristi za grafički opis izgleda stranica na Webu. HTML dokument sastoji se od teksta i HTML oznaka. Oznake opisuju sadržaj pojedinih dijelova dokumenta, kao što su naslovi, paragrafi, tablice, tipografija itd. Osnovna zadaća HTML-a jest opisati sadržaj dokumenta tako da on može biti prikazan uvijek na isti način, neovisno o pregledniku ili operacijskom sustavu. HTML nije programski jezik te njime ne možemo izvršiti neku konkretnu

<sup>2</sup> U trenutku pisanja ovog rada navedena opcija nije dostupna te Slika 2 nije autentična nego je kreirana za potrebe rada.

zadaću. HTML je jednostavan za upotrebu i lako se uči, zbog čega je opće prihvaćen i popularan (Henick, 2010).

HTML dokument se sastoji od osnovnih HTML elemenata. HTML elementi sastoje se od parova HTML oznaka koje se pišu pomoću kutnih zagrada. Oznake kao što su `<img />` i `<input />` izravno uvode sadržaj u web stranicu. Druge oznake, kao `<p> tekst </p>`, okružuju tekst i pružaju informacije o tekstu dokumenta te mogu sadržavati druge oznake kao pod-elemente. Web preglednici ne prikazuju HTML oznake nego ih koriste za interpretaciju sadržaja web stranice (Schmitt, Simpson, 2011).

HTML elementi mogu imati svoje atribute. Većina atributa dolazi u parovima „naziv-vrijednost“ razdvojenim znakom „=“, a pišu se unutar početne oznake elementa. Vrijednost atributa piše se unutar znakova navodnika. Postoji nekoliko čestih atributa koji se pojavljuju u mnogim elementima. To su `id`, `class`, `style`, `title` i `lang`.

U ovom projektu korištena je HTML verzija HTML5. HTML verzija 5 unaprijedila je mogućnosti prikaza najnovije multimedije uvođenjem oznaka `<video>`, `<audio>` i `<canvas>`. Ova nova svojstva omogućuju HTML dokumentima da samostalno učitavaju multimedijske sadržaje, bez pomoći drugih alata kao što je Flash. Također su poboljšane mogućnosti prilagodbe za slabije uređaje kao što su stariji mobilni telefoni i prijenosna računala. (W3C, 2017).

Verzija HTML-a koja je korištena pri izradi stranice određuje se deklaracijom u početku HTML dokumenta, prije prve `<html>` oznake. Za HTML5 ta deklaracija je `<!DOCTYPE html>`. Ova deklaracija nije HTML oznaka, već uputa web pregledniku o tome u kojoj je verziji HTML-a stranica napisana. Elementi HTML verzije 5 korišteni u izradi ovog projekta su `<header>`, `<footer>` i `<nav>`. Kako bi se HTML dokument ispravno prikazao, web preglednik treba znati koji set znakova je korišten u izradi stranice, odnosno kako je stranica kodirana. Kodiranje stranice određeno je u oznaci `<meta>`, a u ovom projektu koristio sam oznaku `<meta charset="UTF-8">`.

## 2.5. CSS

Cascading Style Sheets (CSS) stilski je jezik koji opisuje kako će HTML elementi biti prikazani na ekranu, papiru ili nekom drugom mediju. Jedan CSS stilski list može kontrolirati izgled više web stranica odjednom. CSS stilski list može se pisati unutar HTML dokumenta, a označava se oznakom `<style>`. Vanjski stilski listovi pohranjuju se u CSS datoteke, a u HTML-u se pozivaju oznakom `<link rel="stylesheet" href="stil.css">`.

U ranijim verzijama HTML-a, HTML dokument opisivao je i sadržaj stranice i način na koji sadržaj treba biti prikazan. No kada su u verziju 3.2 HTML-a uključene oznake koje opisuju način prikaza boja, tipografije i drugih elemenata, pisanje HTML dokumenata na ovaj način postalo je dugotrajan i mukotrpan posao, jer je te oznake bilo potrebno uvrstiti u svaki HTML dokument. Kako bi riješili ovaj problem, World Wide Web konzorcij stvorio je CSS. CSS je stvoren s namjerom da odvoji sadržaj od oblikovanja. (W3C, 2017).

CSS, kao ni HTML, nije programski jezik. CSS sintaksa sastoji se od popisa pravila koja opisuju izgled elemenata HTML dokumenta. Svako pravilo ili set pravila sastoji se od jednog ili više selektora i deklaracijskog bloka. Selektori određuju na koji će se element HTML dokumenta primjenjivati stil. Jedan selektor može opisivati izgled više elemenata, ali i više selektora može opisivati izgled jednog elementa (W3C, 2011).

Za izradu ovog projekta korištena je verzija CSS3. Za razliku od CSS2, koji sadrži jednu specifikaciju koja opisuje razna svojstva, CSS3 je podijeljen u više dokumenata koje nazivamo modulima. Svaki modul proširuje mogućnosti definirane u CSS2 ili dodaje nove (Bos, 2017).

## 2.6. Javascript

JavaScript je skriptni programski jezik koji se izvršava u web pregledniku na strani klijenta. Služi kako bi se web stranice učinile interaktivnima. Koristi ga većina web stranica a podržavaju ga svi moderni web preglednici pomoću ugrađenih pokretačkih programa. JavaScript pokretački programi u početku su implementirani samo u web preglednike na strani klijenta, no danas su ugrađeni u više drugih vrsta aplikacija, kao što su poslužiteljske baze podataka, tekstualni procesori ili PDF softver. Prva javno dostupna inačica JavaScripta objavljena je s web preglednikom Netscape Navigator 2.0, 1995. godine (Nixon, 2009).

JavaScript je i dinamički tipizirani jezik, što znači da tip podatka vezemo uz vrijednost, a ne uz izraz. Na primjer, varijabla u jednom trenutku može biti vezana uz broj, a u sljedećem uz riječ. U ovom projektu korišten je Google Maps JavaScript API za izradu interaktivne karte. Google Maps JavaScript API omogućuje prikaz i uređivanje stila karte, a nešto više o njemu reći ću u nastavku rada.

## 2.7. Google Maps API

Google Maps API (engl. *Application Programming Interface*) je sučelje za programiranje namijenjeno integraciji Google Maps aplikacije u web stranice, a dostupno je za web preglednike, mobilne operacijske sustave Android i iOS te putem HTTP web servisa. Korisničko okruženje zasnovano na XML-u omogućuje korisnicima da kreiraju i prilagode vlastite interaktivne karte. U ovom projektu Google Maps API korišten je za izradu interaktivne karte što ću detaljno opisati u poglavlju 4.3. Kreiranje vlastite Google Maps aplikacije zahtjeva autorizaciju pomoću Google Maps API ključa. API ključ je moguće zatražiti nakon prijave na Google korisnički račun. Korištenje ključa besplatno je za izradu aplikacija u nekomercijalne svrhe (Google Developers, 2017).

## 2.8. Bootstrap

Bootstrap je besplatni front-end web okvir (engl. Framework) otvorenog koda za dizajniranje web stranica i web aplikacija. Sastoji se od HTML i CSS predložaka za tipografiju, forme, gumbe, navigaciju i ostale dijelove sučelja. Nudi također i neobavezne JavaScript ekstenzije. Bootstrap su razvili zaposlenici Twittera Mark Otto i Jacob Thornton kako bi olakšali konzistentno korištenje internih alata.

Bootstrap se sastoji od niza stilskih listova koji implementiraju razne dijelove samog alata. Ti stilski listovi uglavnom se kompiliraju u pakete i uključuju u web stranice, ali mogu se koristiti ili izbaciti samo pojedinačni dijelovi. Za izradu ovog projekta korišten je selektor klase „custom-select“, za HTML oznaku `<select>` (Bootstrap, 2017).

### 3. Priprema za izradu web sjedišta

U ovom poglavlju opisat ću pripremu za izradu web sjedišta, od početne motivacije za projekt do izrade baze podataka i strukture web sjedišta. Realizaciju sjedišta detaljno ću opisati u sljedećem poglavlju.

Web mjesto ili web sjedište je mjesto na lokalnom računalu ili web poslužitelju koje se sastoji od više međusobno povezanih web stranica, a priprema se i održava kao jedna informacijska cjelina. Web sjedište ima početnu stranicu koja opisuje njegovu osnovnu funkciju te sadrži poveznice prema njegovim ostalim stranicama (TheFreeDictionary.com, 2016).

#### 3.1. Ideja i svrha projekta

Na izradu ovog projekta motivirao me moj rad u CARNet-ovoj službi Podrške obrazovnom sustavu. U svom radu dobro sam se upoznao s prednostima i nedostacima sustava NISpuSŠ te s potrebama njegovih korisnika.

Ideja projekta bila je izraditi interaktivni informacijski sustav koji će na interaktivnoj karti prikazati dostupne obrazovne programe i lokacije škola u kojima se izvode, prema kriteriju koji odabere korisnik. Dodatno, korisnici mogu vidjeti broj bodova koji je bio potreban za ostvarivanje prava na upis u određeni program u školsku godinu 2017./2018.. Projekt je vizualno baziran na izgledu sustava NISpuSŠ.

Projekt trenutno sadrži informacije samo o školama i obrazovnim programima u gradu Zagrebu. U budućnosti planiram bazu proširiti informacijama o svim javnim školama u Hrvatskoj, kao i o bodovima potrebnima ranijih godina.

Svrha ovog projekta je pružiti učenicima pomoć pri odabiru srednjoškolskih obrazovnih programa za koje se žele prijaviti, tako da im na jednostavan način prikaže

lokacije škola koje izvode odabrani program te broj bodova koji je ranijih godina bio potreban za ostvarivanja prava na upis. Iako je u sustavu NISpuSŠ, na prikazu detalja o programu lokacija škole prikazana na karti, aplikacijom sam htio prikazati lokacije više škola istovremeno. Također moguće je filtrirati prikaz prema vrsti programa te prema nazivu programa, a prikazan je i broj bodova potreban za upis prethodne godine. Navedenim funkcionalnostima htio sam nadomjestiti neke od nedostataka sustava NISpuSŠ.

### 3.2. Izrada baze podataka

Prije izrade web sjedišta potrebna je baza podataka s informacijama koje će se prikazivati na interaktivnoj karti. Popis škola s informacijama o obrazovnim programima i bodovima potrebnim za upis dostupan je na naslovnoj stranici sustava NISpuSŠ. Podatke sam strukturirao u Excel tablici te sam dodao informacije o zemljopisnoj širini i dužini lokacije škola, što je prikazano na slici 3. Podatke o zemljopisnoj širini i dužini ručno sam dohvaćao s Google Mapa za svaku školu pojedinačno.

	A	B	C	D	E	F
1	škola	program	vrsta	bodovi	lat	long
2	Poljoprivredna škola	Agrotehničar - novi strukovni kurikulum	strukovni	45,00	45.832224	16.061832
3	Poljoprivredna škola	Agroturistički tehničar	strukovni	47,21	45.832224	16.061832
4	Industrijska strojarstva škola	Alatničar	strukovni	25,12	45.798741	15.995731
5	Graditeljska tehnička škola	Arhitektonski tehničar	strukovni	60,59	45.772732	15.981937
6	Elektrostrojarstva obrtnička škola	Autoelektričar - JMO	strukovni	23,44	45.804648	15.944346
7	Obrtnička i industrijska graditeljska škola	Autolakirer	strukovni	23,11	45.772918	15.981394
8	Elektrostrojarstva obrtnička škola	Autolimar - JMO	strukovni	21,20	45.804648	15.944346
9	Industrijska strojarstva škola	Autolimar - JMO	strukovni	26,77	45.798741	15.995731
10	Elektrostrojarstva obrtnička škola	Automehaničar - JMO	strukovni	22,35	45.804648	15.944346
11	Industrijska strojarstva škola	Automehaničar - JMO	strukovni	22,06	45.798741	15.995731
12	Elektrostrojarstva obrtnička škola	Automehatroničar - JMO	strukovni	25,05	45.804648	15.944346
13	Industrijska strojarstva škola	Bravar - JMO	strukovni	22,57	45.798741	15.995731
14	Industrijska strojarstva škola	CNC operater	strukovni	25,24	45.798741	15.995731
15	Strojarstva tehnička škola Fausta Vrančića	CNC operater	strukovni	25,52	45.797673	15.995789
16	Poljoprivredna škola	Cvjećar	strukovni	25,61	45.832224	16.061832

Slika 3: Struktura podataka u Excel datoteci

Ovako strukturirane podatke potom sam izvezo u XML datoteku. XML datoteka sastoji se od glavnog čvora (engl. *parent node*) <markers> u kojem su podaci o svakom pojedinačnom obrazovnom programu smješteni u pod-čvor (engl. *child node*) <marker>. Struktura čvora u XML datoteci prikazana je u kodnom bloku 1.

```
<?xml version="1.0"?>
<markers>
  <marker id="1" skola="I. gimnazija" program="Opća
    gimnazija" vrsta="gimnazijski" bodovi="74.00"
    lat="45.777473" lng="15.992900"/>
  <marker id="2" skola="II. gimnazija" program="Opća
    gimnazija" vrsta="gimnazijski" bodovi="79.00"
    lat="45.808472" lng="15.986638"/>
  <marker id="3" skola="III. gimnazija" program="Opća
    gimnazija" vrsta="gimnazijski" bodovi="74.00"
    lat="45.811794" lng="16.008303"/>
</markers>
```

Kodni blok 1: Struktura XML čvora

### 3.3. Struktura web sjedišta

Prije izrade web sjedišta potrebno je odrediti od kojih će se dijelova ono sastojati. U informacijskim sustavima razlikujemo komponente za prikupljanje, obradu, čuvanje i distribuiranje podataka. Ovo web sjedište sastoji se od sljedećih dijelova:

- indeks.html – naslovna stranica web sjedišta koja sadrži osnovne informacije o sjedištu i poveznice na ostale stranice web sjedišta
- izbornik.html – stranica s izbornikom za odabir obrazovnih programa i kartom na kojoj su prikazane lokacije škola koje ih izvode
- izbori.js – JavaScript datoteka koja generira mogućnosti odabira u izborniku
- karta.js – JavaScript datoteka koja generira kartu i prikazuje lokacije škola
- baza.xml – baza podataka u XML datoteci koja sadrži potrebne informacije o školama i obrazovnim programima
- stil.css – stilski list koji sadrži informacije o oblikovanju HTML dokumenata.



## 4. Izrada web sjedišta

U ovom ću poglavlju detaljno opisati postupak izrade interaktivne karte pomoću Google Maps JavaScript API-ja. Objasnit ću na koji su način stranice web sjedišta kodirane HTML-om i CSS-om te na koji način je iskorišten JavaScript.

### 4.1. HTML struktura web sjedišta

Na stranicama web sjedišta postoji zaglavlje. Zaglavlje je smješteno u element `<header>`, a sastoji se nekoliko dijelova koji prikazuju vanjske poveznice, naslov web sjedišta i element navigacijske trake `<nav>`. Element `<nav>` definira set navigacijskih poveznica koje vode prema ostalim stranicama web sjedišta. HTML struktura zaglavlja prikazana je u kodnom bloku 2.

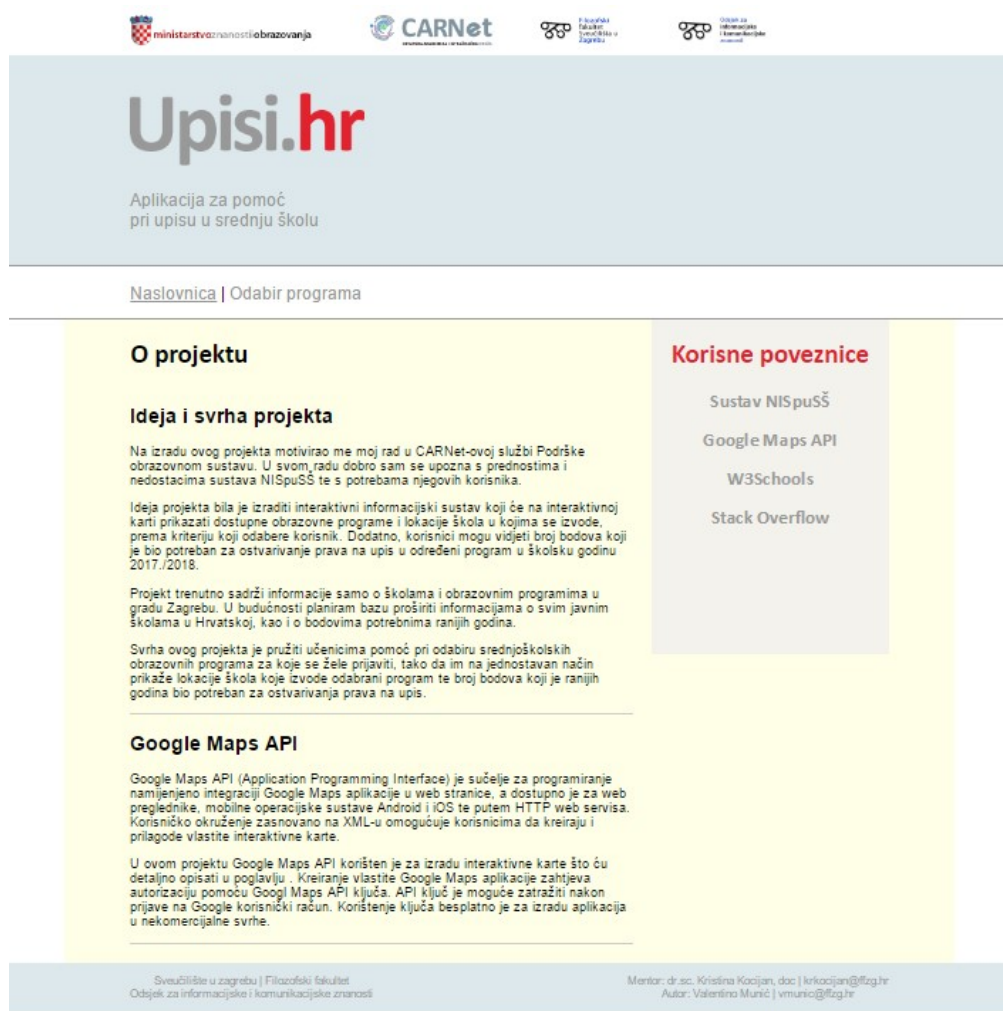
```
<nav id="bar">
  <a class="nav" id="disabled">Naslovnica</a> |
  <a href="izbornik.html" class="nav">Stranica 1</a>
</nav>
```

**Kodni blok 2:** Definiranje navigacijske trake

Osim zaglavlja, svaka stranica web sjedišta ima i podnožje. Podnožje web stranice definirano je elementom `<footer>`. Unutar podnožja, sadržaj je razdvojen u dva elementa `<span>`, kako bi se lakše upravljalo njegovim izgledom. HTML struktura podnožja prikazana je u kodnom bloku 3. Završni izgled naslovne stranice prikazan je na slici 4.

```
<footer>
  <div id="podnozje">
    <span id="ffzg">
      Sveučilište u zagrebu | Filozofski fakultet<br/>
      Odsjek za informacijske i komunikacijske znanosti
    </span>
    <span id="kontakt">
      Mentor: dr.sc. Kristina Kocijan, doc |
      krkocijan@ffzg.hr<br/>
      Autor: Valentino Munić | vmunic@ffzg.hr
    </span>
  </div>
</footer>
```

Kodni blok 3: HTML struktura podnožja



Slika 4: Naslovna stranica web sjedišta

Na stranici koja sadrži kartu, potrebno je definirati elemente izbornika pomoću kojeg ćemo upravljati informacijama koje se na karti prikazuju. U ovom slučaju radi se o padajućem izborniku koji je definiran elementom `<select>`. U izborniku „Odaberi vrstu programa:“ definirana su tri moguća odabira, pomoću elemenata `<option>`. HTML struktura izbornika prikazana je u kodnom bloku 4.

```
<form id="izbornik">
  <table>
    <tr>
      <td align="right">Odaberi vrstu programa:</td>
      <td width="10px"></td>
      <td>
        <select      class="custom-select      select"      id="ddl"
onchange="configureDropDownLists(this,document.getElementById(
'ddl2'));">
          <option hidden></option>
          <option value="sve">Sve vrste programa</option>
          <option              value="gimnazijski">Gimnazijski
programi</option>
          <option              value="strukovni">Strukovni
porgrami</option>
        </select>
      </td>
    </tr>
    <tr>
      <td align="right">Odaberi program:</td>
      <td width="10px"></td>
      <td>
        <select      width="100px"      class="custom-select      select"
id="ddl2"      onchange="initMap(this.value);">
          </select>
        </td>
      </tr>
    </table>
  </form>
```

**Kodni blok 4:** HTML struktura izbornika

U izborniku „Odaberi program:“ mogući odabiri nisu unaprijed definirani, već se generiraju ovisno o odabiru u prvom izborniku. To je postignuto na način da se odabirom opcije u prvom izborniku pokreće JavaScript funkcija `configureDropDownLists()` koja generira mogućnosti odabira u drugom izborniku. Funkcija `configureDropDownLists()` prikazana je u kodnom bloku 5.

```
function configureDropDownLists(ddl1,ddl2) {
    var sve = new Array(
        'popis svih programa'
    );
    switch (ddl1.value) {
        case 'sve':
            ddl2.options.length = 0;
            for (i = 0; i < sve.length; i++) {
                createOption(ddl2, sve[i], sve[i]);
            }
            break;
    }
}

function createOption(ddl, text, value) {
    var opt = document.createElement('option');
    opt.value = value;
    opt.text = text;
    ddl.options.add(opt);
}
```

**Kodni blok 5:** Funkcija `configureDropDownLists()`

U funkciji su zadana tri polja, po jedno za svaku mogućnost odabira u prvom padajućem izborniku. Funkcija na temelju odabira iz prvog izbornika iterira kroz odgovarajuće polje i generira mogućnosti odabira za drugi izbornik. Funkcija `createOption()` zatim generirane mogućnosti učitava u drugi izbornik.

Funkcije `configureDropDownLists()` i `createOption()` pohranjene su u vanjskoj JavaScript datoteci `izbori.js`. Pohranjivanje JavaScript koda u vanjsku datoteku čini HTML dokument urednijim i jednostavnijim za uređivanje. Datoteka `izbori.js` učitana je u HTML dokument elementom pomoću `<script type="text/javascript" src="karta.js">`.

## 4.2. CSS struktura web sjedišta

Za ovo web sjedište korišten je vanjski CSS stilski list, pohranjen u datoteci `stil.css`. CSS datoteka je u HTML dokument učitana pomoću elementa `<link rel="stylesheet" href="css/stil.css">`. Korištenje vanjske CSS

datoteke, kao i kôd JavaScripta, čini HTML dokument urednijim i jednostavnijim za korištenje. Također, izmjenom jedne CSS datoteke možemo utjecati na sve stranice web sjedišta.

U CSS dokumentu opisan je način prikaza pojedinih elemenata, ali i elemenata koji su zajednički svim stranicama web sjedišta. Tako je navigacijska traka opisana selektorima `.nav`, `#linkovi` i `#disabled`. Element `<div>` s atributom `„id='linkovi'“` određuje položaj i veličinu navigacijske trake. Atributom `„id='disabled'“` označena je poveznica stranice na kojoj se trenutno nalazimo te se ona neće ponašati kao poveznica. CSS struktura navigacijske trake prikazana je u kodnom bloku 6

```
.nav {
    color: #999999;
    text-decoration: none;
    display: inline-block;
}
.nav:hover {
    text-decoration: underline;
}
#linkovi {
    width: 100%;
    height: 52px;
    border-top: solid 2px #999999;
    border-bottom: solid 2px #999999;
}
#disabled {
    cursor: default;
    text-decoration: underline;
}
```

**Kodni blok 6:** CSS oblikovanje navigacijske trake

Selektorom `.nav` određena je boja i oblik teksta poveznica u navigacijskoj traci. Svojstvom `display: inline-block;` određeno je da se poveznice u navigacijskoj traci učitavaju vodoravno jedna do druge.

Padajući izbornici na stranici „Odabir programa“ uređeni su pomoću Bootstrap selektora `.custom-select`. CSS struktura selektora `.custom-select` prikazana je u kodnom bloku 7.

```
.custom-select {
    display: inline-block;
    max-width: 100%;
    height: calc(2.25rem + 2px);
    padding: 0.375rem 1.75rem 0.375rem 0.75rem;
    line-height: 1.25;
    color: #495057;
    vertical-align: middle;
    background: #fff url("url") no-repeat right 0.75rem
    center;
    background-size: 8px 10px;
    border: 1px solid rgba(0, 0, 0, 0.15);
    border-radius: 0.25rem;
    -webkit-appearance: none;
    -moz-appearance: none;
    appearance: none;
}
```

**Kodni blok 7:** Struktura Bootstrap selektora `.custom-select`

### 4.3. Kreiranje Google Maps JavaScript karte

Za učitavanje karte potrebno je odrediti jedan element `<div>` s atributom „id=map“. Tom elementu potrebno je CSS-om odrediti dimenzije, što određuje dimenzije same karte. Karta je učitana pomoću vanjske JavaScript datoteke, elementom `<script src="https://maps.googleapis.com/maps/api/js?key = APIključ &callback=initMap" async defer>`. Kako sam ranije naveo, za korištenje Google Maps JavaScript API-ja potrebno je zatražiti vlastiti API ključ. Atributi `async` `defer` određuju da će se karta učitati tek nakon što se učita cijeli HTML kod stranice.

Kartu je također moguće prilagoditi određenim željama. Za opcije prikaza i prilagodbe karte potrebno je opet kreirati vanjsku JavaScript datoteku u koju ćemo pohraniti funkcije prilagodbe karte i učitavanja lokacija. Datoteka je u HTML dokument učitana pomoću elementa `<script type="text/javascript" src="karta.js">`. Ova karta prilagođena je na način da je početni stupanj uvećanja karte 12, prikazana je kao cestovna mapa a u centru karte prikazan je centar Zagreba. Na karti su sakrivene uobičajene mogućnosti odabira vrste prikaza karte. Također, sve točke interesa, osim škola, na karti su sakrivene. To je postignuto funkcijom `styles`, a njen primjer prikazan je u kodnom bloku 8.

```
function initMap() {  
    var map = new  
    google.maps.Map(document.getElementById('map'), {  
        center: new google.maps.LatLng(45.811363,  
15.976828),  
        zoom: 12,  
        disableDefaultUI: true,  
        zoomControl: true,  
        styles: [  
            {  
                "featureType": "administrative",  
                "elementType": "geometry",  
                "stylers": [ { "visibility": "off" } ]  
            },  
            {  
                "featureType": "poi.school",  
                "stylers": [ { "visibility": "on" } ]  
            }  
        ]  
    })  
}
```

**Kodni blok 8:** Učitavanje i prilagodba karte

Odabirom opcije iz drugog padajućeg izbornika na karti se učitavaju lokacije škola. Funkcija `downloadUrl()` učitava XML datoteku `baza.xml`, koja sadrži informacije o školama i obrazovnim programima. Funkcija potom deklarira polje koje popunjava podacima iz XML datoteke. Funkcija `downloadUrl()` prikazana je u kodnom bloku 9.

```
downloadUrl('baza.xml', function(data) {
    var xml = data.responseXML;
    var markers =
    xml.documentElement.getElementsByTagName('marker');
    Array.prototype.forEach.call(markers,
    function(markerElem) {
        var id = markerElem.getAttribute('id');
        var skola = markerElem.getAttribute('skola');
        var program = markerElem.getAttribute('program');
        var vrsta = markerElem.getAttribute('vrsta');
        var bodovi = 'Bodovi potrebni za upis: ' +
        markerElem.getAttribute('bodovi');
        var point = new google.maps.LatLng(
        parseFloat(markerElem.getAttribute('lat')),
        parseFloat(markerElem.getAttribute('lng')));
    });
});
```

**Kodni blok 9:** Funkcija `downloadUrl()`

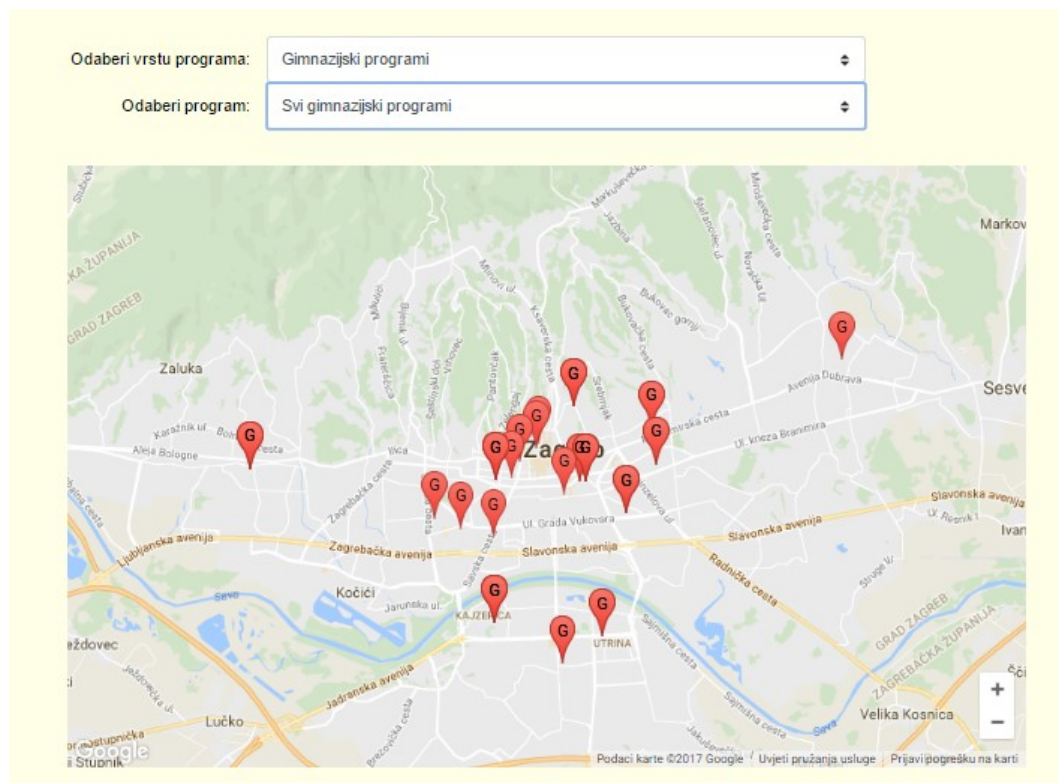
Funkcija zatim iterira kroz polje pomoću petlje „for“ i prikazuje na karti one lokacije koje se podudaraju s odabirom u padajućem izborniku. Primjerice, ako je u padajućem izborniku odabrana opcija „Svi programi“, funkcija neće raditi provjeru nego će iterirajući kroz polje na karti prikazati lokacije svih škola. Ako je odabrana opcija „Svi strukovni programi“, funkcija će na karti prikazati samo one programe kod kojih je vrijednost varijable `vrsta` jednaka izrazu „strukovni“.

Lokacije na karti imaju oznaku „G“ za gimnazijske programe, odnosno „S“ za strukovne. Klikom na oznaku lokacije otvorit će se prozorčić s informacijama o programu: imenom škole, nazivom programa i brojem bodova potrebnih za upis. Dio koda funkcije `downloadUrl()` koji na karti prikazuje lokacije prikazan je u kodnom bloku 10. Završni izgled karte prikazan je na slici 5.



```
var customLabel = {  gimnazijski: { label: 'G' },
                     strukovni: { label: 'S' }
                   };
var infowincontent = document.createElement('div');
var strong = document.createElement('strong');
               strong.innerHTML = skola
               infowincontent.appendChild(strong);
infowincontent.appendChild(document.createElement('br'));
var text = document.createElement('text');
text.innerHTML = program + '<br/>' + bodovi
infowincontent.appendChild(text);
var icon = customLabel[vrsta] || {};
var marker = new google.maps.Marker({
  map: map,
  position: point,
  label: icon.label
});
marker.addListener('click', function() {
  infoWindow.setContent(infowincontent);
  infoWindow.open(map, marker);
});
```

**Kodni blok 10:** Dio funkcije koji učitava oznake na karti



Slika 5: Završni izgled karte

## 5. Zaključak

U ovom radu opisao sam postupak izrade informacijskog sustava u mrežnom okruženju, na primjeru izrade interaktivne karte. Postupak izrade opisao sam od početne ideje projekta, preko pripreme za izradu web sjedišta, do samog projektiranja aplikacije.

Ovaj projekt baziran je na Nacionalnom informacijskom sustavu prijava i upisa u srednje škole, čije sam prednosti i nedostatke detaljno objasnio. Kako sam naveo, svrha ovog projekta je nadomjestiti neke nedostatke sustava NISpuSŠ i olakšati postupak upisa u srednje škole. Osim sustava NISpuSŠ, naveo sam i opisao neke opće pojmove i alate koji su korišteni pri izradi ovog projekta.

Veći dio rada posvetio sam samom postupku izrade projekta. Opisao sam strukturu web sjedišta te način na koji su web stranice kreirane. Glavna karakteristika ovog rada su izbornik i interaktivna karta, koje sam detaljno opisao.

Ovaj projekt može poslužiti kao pomoć učenicima pri odabiru programa za upis u srednju školu. Projekt također može poslužiti za izradu statistike broja bodova potrebnog za upis u jednu vrstu programa u različitim školama i slično. U budućnosti projekt planiram proširiti informacijama o svim javnim školama u Hrvatskoj te predložiti njegovu integraciju u sustav NISpuSŠ.

## 6. Literatura

1. Bootstrap, „About“, <http://getbootstrap.com/docs/3.3/about/>, (posjećeno: 19.9.2017.), 2017.
2. Bos, B., „CSS Specifications“, <https://www.w3.org/Style/CSS/specs>, (posjećeno: 19.9.2017.), 2017.
3. CARNet, „Društveni softver: Web 2.0“, <http://www.carnet.hr/tematski/drustvenisoftver/pojmovnik>, (posjećeno: 18.9.2017.), 2009.
4. Google Maps API Documentation, <https://developers.google.com/maps/documentation/>, (posjećeno: 19.9.2017.), 2017.
5. Henick, B., „HTML & CSS: The Good Parts“, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, 2010. Str: 7-24.
6. Ministarstvo znanosti i obrazovanja, „Procedura upisa – NISpuSŠ“, <https://mzo.hr/hr/rubrike/procedura-upisa-nispuss>, (posjećeno: 18.9.2017.), 2017.
7. Nixon, R., „Learning PHP, MySQL, and JavaScript“, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, 2009. Str: 337-419.
8. Radovan, M., „Projektiranje informacijskih sustava“, Informator, Zagreb, 1991.
9. Schmitt, C., Simpson K., „HTML5 Cookbook“, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, 2012. Str: 1-27.
10. The Free Dictionary, „Website“, <http://www.thefreedictionary.com/Website>, (posjećeno: 22.9.2017.), 2016.
11. Tuđman M., Boras D., Dovedan Z., „Uvod u informacijske znanosti“, pog. 9.1. „Odrednice informacijskih sustava“, <http://www.ffzg.unizg.hr/infoz/dzs/text/Uvod%20u%20informacijske%20znanosti/pog9.htm#91>, (posjećeno 18.9.2017.), Zagreb, 1991.

12. W3C, „HTML5“, <https://www.w3.org/TR/html5/>, (posjećeno: 26.9.2017.), 2014.
13. W3C, „Selectors Level 3“, <https://www.w3.org/TR/selectors/>, (posjećeno: 22.9.2017.)